RS synchronized time

Synchronized time in RS? 1(3)

- RS needs to determine if the authorization information is valid
- Question on the ACE list if we only consider time-based freshness
- Continue discussion today to move this question forward

Background

- The access token may contain an expiry time claim ("exp”)
- To use this claim RS needs to have synchronized time with AS
- Can we assume that RS always have synchronized time?

Synchronized time in RS? 2(3)

- Token introspection allows for validation of authz info
- But RS can’t always connect to AS (RS ”offline") (6.1, 6.2, 6.5)
  - No connectivity (e.g. container monitoring use case)
  - Not affordable to have constant connectivity (e.g. energy, cost)

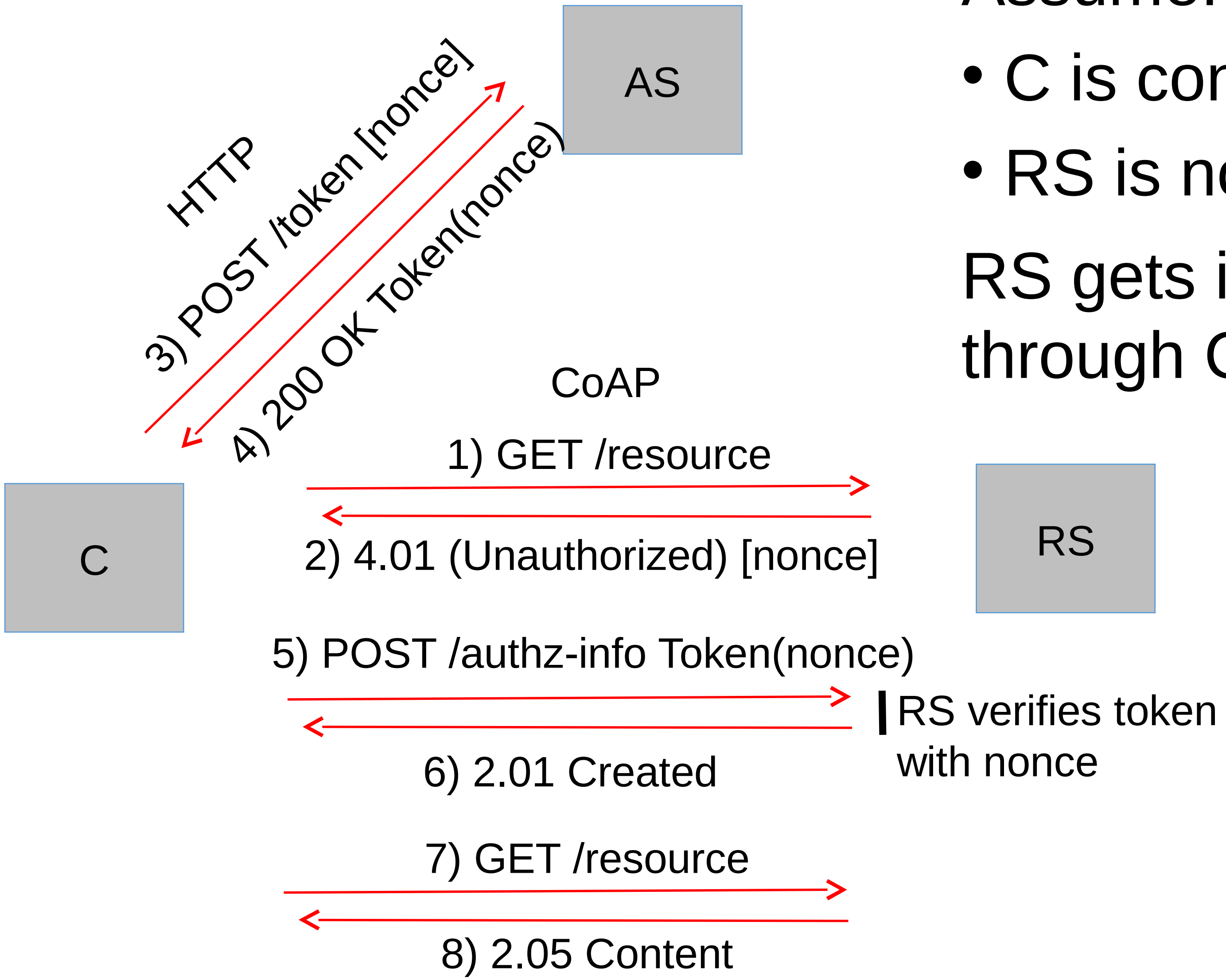
- When RS is ”offline”, how does it synchronize time?
- When RS is ”offline”, how does it check revokation?
- How does RS even know that it needs to synchronize time/perform revokation unless it checks?

Synchronized time in RS? 3(3)

Possible assumptions:

1. RS is never offline
   - how do we handle those use cases/deployment scenarios?
2. RS neither synchronizes time nor checks revocation while being offline
   - what if RS is offline for a long time period?
3. RS optionally verifies validity through C
   - A simple example on next slide. See mail thread for other examples.
   - Would this be a one-time-token, or can the device rely on internal clock?

Example: Nonce-based freshness



AS

C

RS

HTTP

3) POST /token [nonce]

4) 200 OK Token(nonce)

CoAP

1) GET /resource

2) 4.01 (Unauthorized) [nonce]

5) POST /authz-info Token(nonce)

6) 2.01 Created

RS verifies token with nonce

7) GET /resource

8) 2.05 Content

Assume:

- C is connected to AS
- RS is not connected to AS

RS gets information about validity through C (non-time based freshnes